

Local reasoning for a distributed synchronous data-centric algorithm

Jean-François Monin

Proving that a system made of several components satisfies desired properties remains a very hard challenge to the scientific community, even with the help of a proof-assistant. This is clear, for instance, on the history of a well-studied distributed algorithm for computing a minimum-weight spanning tree, due to Gallager, Humblet and Spira [GHS83]. The rigorous proofs made between 1987 and 2006 [WLL88, Hes99, MS06] are all intricate and very long (100 to 170 pages).

We focus here on systems which aim at building a distributed data-structure enjoying global properties, e.g., topological properties. A common example is given by routing tables. The challenge is to find suitable invariants. Even for simpler algorithms, such as BFS [DGM11], PRIM or election, reasoning are very subtle and clearly, automated proof checking is required if we want to ensure that real systems behave as expected.

Several approaches were developed in the last decades, but there is still much room for improvement. On the other hand, a recent trend in verification called *local reasoning* [PRZ01] seems quite promising. Complete local reasoning algorithms have been developed for safety properties, linear properties and properties with fairness constraints.

In our opinion, the point is not only that local reasoning extends the capacities of automated techniques but, more importantly, that it provides deep insights on the design of distributed algorithms.

We illustrate this idea on a program which constructs a spanning tree over a distributed synchronous system. Our framework is Netlog, a language protocols for which we started to develop a proof environment based on Coq. A first experiment, related in [DGM11], provides a correctness proof for a distributed algorithm which builds a breadth-first spanning tree over a distributed network. We present here a new version of this result – fully formalized and proof-checked by Coq as well, where we emphasize locality of arguments and we discuss the advantages and drawbacks of the new version with relation to the previous one.

Just as a short illustration, here are the two versions of a central part of the proof. In the model considered for distributed computations, we consider a graph of nodes containing a database made of datalog facts, and running datalog rules for deriving new facts to be stored or sent to a neighbor. The

presence of a fact φ in a database d is denoted by $\varphi \in d$. The database of facts stored at node loc is denoted by $|loc|$. Similarly, the database of facts arriving a node y from node x is denoted by $|x \rightarrow y|$.

Definition 1. An edge $x \rightarrow y$ is good in a given configuration if and only if, if $onST(x)$ is stored at x , then $onST(y)$ is stored at y or $onST(x)$ is arriving at y . A node y satisfies good-to if and only if all edges $x \rightarrow y$ are good. A configuration satisfies all-good if and only if all its edges are good.

Definition 2 (already stored). An edge $x \rightarrow y$ satisfies ASe iff, if $onST(z) \in |x \rightarrow y|$, then $z = x$ and $onST(z) \in |x|$. A node satisfies ASto iff all edges $x \rightarrow y$ satisfy ASe. A configuration satisfies ASg iff all its edges satisfy ASe.

Proposition 3. $ASg, all-good \xrightarrow{sr} all-good$

Here is the local version of the lemma. We can see that is is more accurate and logically stronger than proposition 3.

Proposition 4. $ASto(y), good-to(y) \xrightarrow{sr} good-to(y)$

References

- [DGM11] Yuxin Deng, Stéphane Grumbach, and Jean-François Monin. A Framework for Verifying Data-Centric Protocols. In R. Bruni and J. Dingel, editors, *FMOODS-FORTE'11*, LNCS, Reykjavik, Iceland, June 6-9 2011. Springer.
- [GHS83] Robert G. Gallager, Pierre A. Humblet, and Philip M. Spira. A Distributed Algorithm for Minimum-Weight Spanning Trees. *ACM Trans. Program. Lang. Syst.*, 5(1):66–77, 1983.
- [Hes99] Wim H. Hesselink. The Verified Incremental Design of a Distributed Spanning Tree Algorithm: Extended Abstract. *Formal Asp. Comput.*, 11(1):45–55, 1999.
- [MS06] Yoram Moses and Benny Shimony. A New Proof of the GHS Minimum Spanning Tree Algorithm. In Shlomi Dolev, editor, *DISC*, volume 4167 of *LNCS*, pages 120–135. Springer, 2006.
- [PRZ01] Amir Pnueli, Sitvanit Ruah, and Lenore Zuck. Automatic deductive verification with invisible invariants. In Tiziana Margaria and Wang Yi, editors, *TACAS*, volume 2031 of *LNCS*, pages 82–97. Springer, 2001.
- [WLL88] Jennifer L. Welch, Leslie Lamport, and Nancy Lynch. A lattice-structured proof of a minimum spanning. In *Proceedings of the seventh annual ACM Symposium on Principles of distributed computing*, PODC '88, pages 28–43, New York, NY, USA, 1988. ACM.